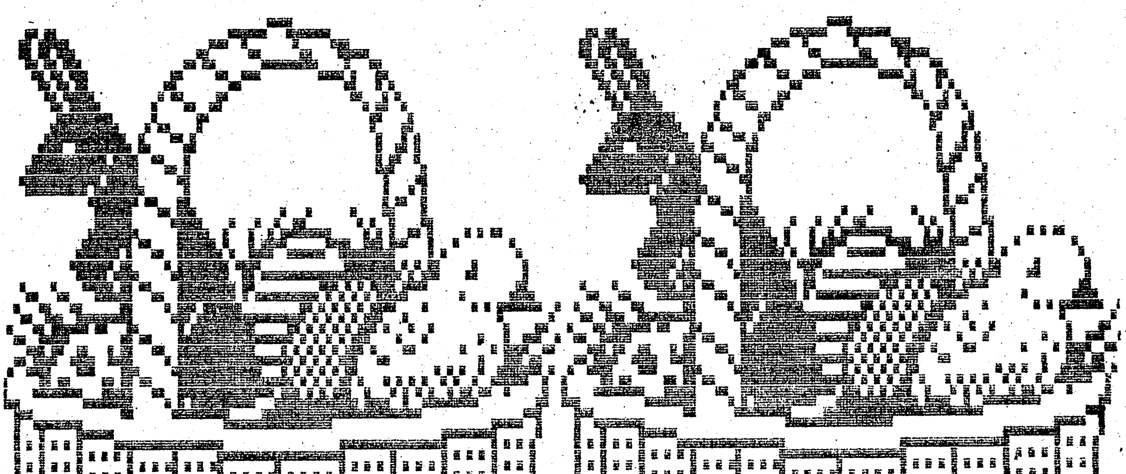
MARCH/APRIL 1988 ISSUE # 17

Hunter Valley VZ Journal



PRODUCED BI-MONTHLY BY H.V.VZ.U.G

BELIEVE IT OR NOT :-

The Committee of Hunter Valley VZ Users' Goup wishes all our members and their families a happy EASTER. We also hope that the Easter Bunny did'nt miss any of you with his goodies.

NEWCASTLE MICROCOMPUTING EXHIBITION :-
Hunter Valley VZ Users' Group has been invited to participitate in above annual show and barring unforseen circumstances we will be there again this year.

VENUE - HUNTER INSTITUTE OF HIGHER EDUCATION
Formerly known as COLLEGE OF ADVANCED EDUCATION.

DATES AND TIMES - THURSDAY, APRIL 21 - 12 P.M. to 9 P.M.
FRIDAY , APRIL 22 - 9 A.M. to 5 P.M.
SATURDAY, APRIL 23 - 9 A.M. to 5 P.M.

VZ USER GROUPS AND PUBLICATIONS

VZ USER MARK HARWOOD P.O.BOX 154 DURAL N.S.W. 2158

LE'VZ OOP J.C.E. D'ALTON 39 AGNES St. TOOWONG QLND. 4066
VSOFTWAREZ - SOFTWARE/HARDWARE FOR SALE

VZ DOWN UNDER SCOTT LE BRUN 59 BRENTWOOD DVE WANTIRNA 3152
SOFTWARE/HARDWARE FOR SALE

VZ-LINK - PETER J. HILL P.O.BOX 1972 C.P.O. AUCKLAND N.Z.

WAVZ - GRAEME BYWATER P.O. BOX 388, MORLEY W.A. 6062

BRISBANE VZ USERS WORKSHOP - C/O 63 TINGALPA ST. WYNUM WEST 4178

HUNTER VALLEY VZ USERS' GROUP - P.O. BOX 161 JESMOND N.S.W. 2299
EDITOR-JOE LEON (049)51 2756 - SECRETARY-ROSS WOODS (049)71 2843

SUBSCRIPTION - H.V.VZ.JOURNAL - 6 MONTHS $9.00 - 12 MONTHS $18.00
New Zealand - 6 MONTHS $12.00 - 12 MONTHS $24.00

NEW VENUE - NEW DATES - NEW VENUE - NEW DATES - NEW VENUE

MEETINGS - FIRST FRIDAY of MONTH at JESMOND NEIGHBOURHOOD CENTRE
MORDUE PARADE - REAR STOCKLAND MALL (BIG W) JESMOND

NOTE :- When writing to any above or H.V.VZ. Users' Group for information please enclose a S.S.A.E. or NZ 2 Int. Reply Coupons.

Besides my VZ's I also have a Commodore C64 with disk drive. For the C64 there's an overabundance of utilities including disk menus. I was pleased to see one finally make an appearance for the VZ and it is the subject for this review.

To use the DISK MENU is simplicity itself. Just BRUN"MENU" and the disk DIRECTORY is loaded into memory and appears on the centre of the screen in a single column. By using the UP/DOWN ARROW keys it is possible to step UP/DOWN through the directory. The cursor will INVERSE each file as it passes over it.

Only a maximum of 16 files will be displayed at any one time. If your disk has more files than that then simply keep scrolling down till rest of directory is displayed.

By pressing the RETURN key a 'T' or 'B' type file will be LOADed and RUN. Other types will be ignored like DATA, WORD PROCESSOR and SOURCE CODE files as will any 'B' files which load into VIDEO RAM like HI-RES screens.

If a basic program is loaded and BREAK key is used then it's possible to return to the MENU by simply typing in MENU and pressing RETURN. There's no need to reload from disk again. It's a nice and thoughtful feature.

When MENU is first BRUN or recalled from basic the first file at top of screen shown is called NEWDISK. By pressing RETURN over NEWDISK a new directory is loaded into memory. Therefore there's no need to SAVE MENU to each disk as the NEWDISK function allows you to inspect many directories before finally chosing program to LOAD and RUN.

I've been using MENU for a few months now and found it very handy and easy to use. There are two improvements that I'ld like to see done to it and that is an INVERSE function of screen using the INVERSE key and a QUIT option using the (-) key.

INVERSE - If a BINARY program is loaded then you're stuck in a light green screen which is the start up screen of MENU and my preference is the inverse of it.

QUIT - If after BRUNning MENU you change your mind you have to load basic file or switch OFF/ON VZ to regain control.

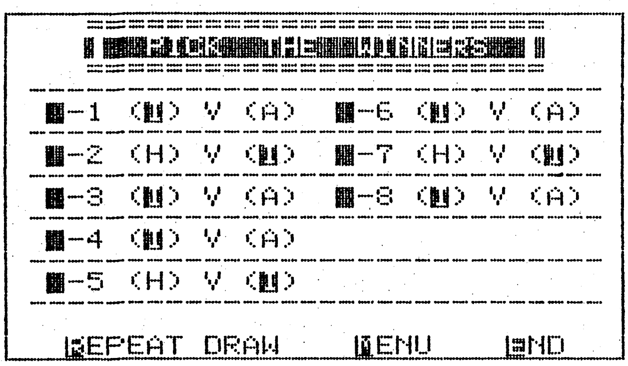
PLEASE NOTE :- The above two comments are not criticisms, but suggestions hopefully leading to improvements and are my own personal opinions.

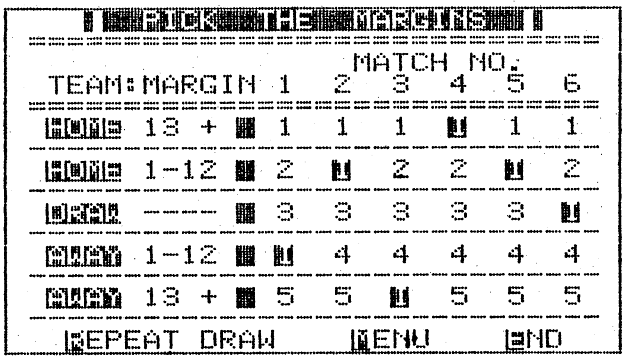
Like all the utilities that were written by Larry TAYLOR and I've had the pleasure of using, MENU follows the same high standard and ease of use we have come to expect from him. I have no hesitation in recommending this program to disk users.

The DISK MENU program is available from :-

Brisbane VZ Users Workshop - C/O 63 Tingalpa St. Wynum West 4178
PRICE $10.00 - Please check for correct price before purchasing.

NOTE :- In the Hunter Valley VZ users can contact Joe LEON for a demonstration of DISK MENU at club meetings.






```
10 GOTO60
50 CLS:RETURN
60 F1$=" ------------------------------":REM 30 (-) SYMBOLS
70 POKE30862,80:POKE30863,52:COLOR,0:POKE30744,96
90 :
100 CLS:POKE30777,1:COLOR7:PRINT@4,"█ NSW VZ FOOTY TAB █"
110 RESTORE:FORPL=32TO63
120 PRINT@PL,"=";:PRINT@PL+192,"=";:NEXT
130 PRINT@103,"█W█  FOR PICK THE WINNER"
140 PRINT@129,"PRESS"
150 PRINT@167,"█M█  FOR PICK THE MARGIN":SOUND20,1;30,1;20,1
160 A$="":A$=INKEY$:A$=INKEY$:IFA$=""THEN160ELSEX=USR(X)
170 IFA$="W"THEN300'WINNER
180 IFA$="M"THEN500'MARGIN
190 GOTO160
290 :
300 FL=0:CLS:POKE30777,1:PRINT@36,"█ PICK THE WINNERS █"
310 COLOR3:FORI=4TO27:PRINT@I,"=":PRINT@I+64,"=":NEXT
320 PRINT@128,"  █-1 (H) V (A)  █-6 (H) V (A)":PRINT
330 PRINT"  █-2 (H) V (A)  █-7 (H) V (A)":PRINT
340 PRINT"  █-3 (H) V (A)  █-8 (H) V (A)":PRINT
350 PRINT"  █-4 (H) V (A)":PRINT
360 PRINT"  █-5 (H) V (A)":PRINT:IFFL=1THEN400ELSEFORI=97TO126
370 PRINT@I,"-":PRINT@I+64,"-":PRINT@I+128,"-":PRINT@I+192,"-"
380 PRINT@I+256,"-":PRINT@I+320,"-":X=USR(X):NEXT
390 SOUND30,1:GOSUB900'"█PRESS SPACE█
400 POKE31477,134:GOSUB50:SOUND0,4:COLOR,1:FOR I=0 TO 7
410 P=134+I*64:IFI>4 THEN P=149+(I-5)*64
420 A$(1)="(H)":A$(2)="(A)
430 RN=RND(2):A$(RN)="(█)"
440 PRINT@P,"            ":X=USR(X):PRINT@P,A$(1);" V ";A$(2);
450 X=USR(X):NEXT
460 POKE31477,132:GOSUB800'"█SUB MENU█
470 FL=1:COLOR3:GOTO320
490 :
500 CLS:POKE30777,1:PRINT@4,"█ PICK THE MARGINS █"
510 COLOR3:PRINT@82,"MATCH NO."
520 PRINT@98,"TEAM:MARGIN 1  2  3  4  5  6":
530 FORI=33TO62:PRINT@I,"=":PRINT@I+96,"=":NEXT
540 FORI=1TO5:READD$:PRINTD$;
550 FORX=1TO6:PRINTI;:X=USR(X):NEXT
560 PRINT:PRINTF1$:NEXT
570 GOSUB900'"█PRESS SPACE█
580 GOTO700
590 COLOR,1:FORX=1TO16STEP3
600 FORI=1TO5
610 B$(I)=CHR$(I+48)
620 NEXT
```

```
630 SL=RND(5)
640 B$(SL)="█"
650 FORY=173TO446STEP64
660 Z=INT(Y/64)-1
670 PRINT@X+Y,B$(Z)
680 NEXT:SOUND25,1:NEXT
690 POKE31477,132:GOSUB800'"█████████"
700 POKE31477,134:GOSUB50:SP=161:FP=449:S=13:F=29:ST=3
710 FORA=SP TO FP STEP64
720 FORB=S TO F STEP ST
730 PRINT@A+B,"?";
740 X=USR(X)
750 NEXT:NEXT
760 SOUND0,4:GOTO590
790 :
800 PRINT@482," REPEAT DRAW    MENU    END  ";:COLOR,0
815 POKE30777,35:SOUND25,1
820 A$="":A$=INKEY$:A$=INKEY$:IFA$=""THEN820ELSEX=USR(X)
830 IF A$="R"THENRETURN
840 IF A$="M"THEN100
850 IF A$="E"THEN:CLS:END
860 GOTO820
890 :
900 COLOR7:PRINT@482,"█████████████████████████████";:SOUND25,1
910 I$="":I$=INKEY$:I$=INKEY$:IFI$=""THEN910ELSEX=USR(X)
920 IFI$<>" "THEN910ELSERETURN
930 :
950 DATA "  HOME 13 + █","  HOME 1-12 █","  DRAW ---- █"
960 DATA "  AWAY 1-12 █","  AWAY 13 + █"
990 :
991 '"*****************************************************
992 '"*   FOOTY  TAB  SELECTOR  WRITTEN  BY  PAUL  LEON    *
993 '"* FOR HIS DAD IN JULY 1984 AND UPDATED IN MARCH 1988 *
994 '"*****************************************************
995 '"*  NOTE :- USE SHIFT 'J' '█' FOR GRAPHIC CHARACTERS  *
996 '"*  IN LINES 320 TO 360 AND LINES 950 TO 960 AS WELL  *
997 '"*****************************************************
998 '"* NOTE - THERE'S NO NEED TO TYPE IN LINES 991 TO 999 *
999 '"*****************************************************
```

With the RUGBY LEAGUE season in full swing punters might find this Footy Tab Selector handy. Besides trying to pick a winner you should also be able to learn a bit about programming. Paul used a few fancy screen handling tricks which you may like to try in your own programs.

This program is menu driven and requires no instructions. When typing in enter line 10 and 50 exactly as shown. It has to do with TOKENS and the RANDOM function and lines 50, 400, 460, 690 and 700.

You'll notice line 50 is unusual, it reads 50 CLS:RETURN, then if we look at line 400 we see POKE31477,134:GOSUB50. If we PEEK at address 31477 we should get 132 which is the TOKEN for CLS and is in line 50. Line 400 POKEs 134 which is the TOKEN for RANDOM and replaces CLS in line 50.

RANDOM makes the numbers more random and the POKE in line 460 replaces RANDOM with CLS in line 50 after selecting the draw. Because RANDOM is a function not normally available from the keyboard the above approach was adopted.

MODIFIED BY DAVE MITCHELL

This is quite a usefull routine. It was first written for the TRS-80 by MR.R.Bowler in 1981 and modified by MR.L.Villani for the VZ200+16k in 1984 and printed in VZ NEWS. In 1987 I modified the existing routine to relocate itself at the top of memory. It enables the User to print the decimal equivalent of hex numbers, Poke and Peek using hex and for adding, subtracting, multipy and divide with a mixture of hex and hex or hex and decimal numbers.

The command '&' is a TRS-80 command that can be used in the VZ. TRY THIS :- PRINT& or POKE& - When RETURN is pressed you will be greeted by this error.

?DISK COMMAND? SYNTAX ERROR

> ERROR

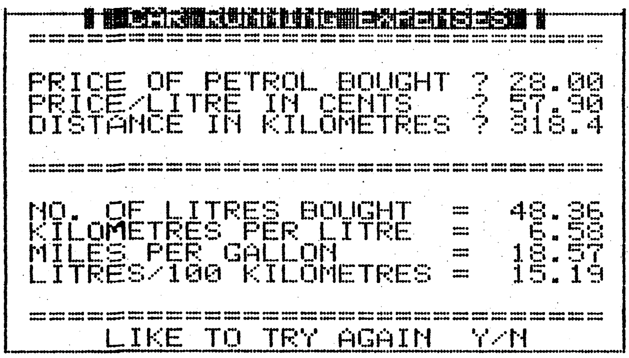
The address for the '&' command is in RAM and can be easily changed to suit your requirements. The command to enable the routine is '&H'.

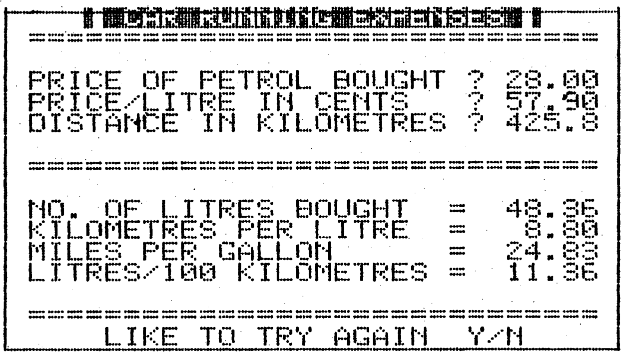
EXAMPLE PRINT&H 7000 will return 28672.
PRINT&H7000+&H41 or PRINT&H7000+65 both will return 28737.
PRINTPEEK(&H7000) will PEEK the address 28672.
POKE&H7000,&H41 will POKE the letter 'A' into address 28672 at the top left side of the screen.

LINE 10 The POKE 30777,1 speeds up the message (lines 10-30 prints the sign on message).
40 Finds the top of memory minus 59 bytes.
50 Sets the top of memory to the new top.
60 Clears 50 bytes of memory for strings and resets 30777.
70 Sets the address for the command '&H'.
90-100 Places the machine code into the addresses above the top of memory.
110 The two pokes is the same as NEW except the sign on message is left on the screen.
120-150 Decimal equivalent of the machine code routine in DATA statments.

The basic program will NEW itself when run. AVOID rerunning the program without resetting the VZ as this will lower the top of memory each time the program is run.

```
2 '***************************************************************
4 '*  HEX CONSTANTS MODIFIED FOR THE VZ IN 1987 BY D.MITCHELL *
6 '***************************************************************
8 :
10 POKE30777,1:CLS:PRINT"      TOP OF MEMORY SEEKING"
20 PRINT"           HEX CONSTANTS":PRINT
30 PRINT" MODIFIED FROM VZ NEWS ISSUE #5  NOVEMBER 1984":PRINT
40 T=PEEK(30897)+256*PEEK(30898)-59
50 POKE30897,T-(256*INT(T/256)):POKE30898,INT(T/256)
60 CLEAR50:POKE30777,35
70 POKE31125,PEEK(30897)+1:POKE31126,PEEK(30898)
80 T%=PEEK(30897)+256*PEEK(30898)-65535
90 FORI%=1TO59
100 READA%:POKET%,A%:T%=T%+1:NEXT
110 POKE31465,0:POKE31466,0:END
120 DATA 215,207,72,6,4,17,0,126,254,48,56,31,254,58,56,10
130 DATA 254,65,56,23,254,71,48,19,214,7,230,15,72,6,4,203,35
140 DATA 203,18,16,250,179,95,65,215,16,220,120,254,4,210,151
150 DATA 25,237,83,33,121,62,2,50,175,120,201
```






```
10 '**********************************************************
20 '*   CAR RUNNING EXPENSES WAS CONVERTED FOR THE VZ 200    *
30 '*   AND VZ 300 FROM A MICROBEE PROGRAM BOOK BY J.LEON    *
40 '**********************************************************
50 :
60 MI$=" ==============================":REM 30 EQUAL SYMBOLS
70 SP$="                                ":REM 32 SPACES
80 F1$="###.##"
90 :
100 CLS:COLOR,0:POKE30744,96
110 COLOR7:PRINT@4,"CAR RUNNING EXPENSES"
120 PRINTMI$:PRINT:SOUND25,1
130 PRINT@96,;:INPUT" PRICE OF PETROL BOUGHT ";PO
140 IFPO=0ORPO>99THENPRINT@96,SP$:GOTO130
150 PRINT@128,;:INPUT" PRICE/LITRE IN CENTS   ";CO
160 IFCO=0ORCO>99THENPRINT@128,SP$:GOTO150
170 PRINT@160,;:INPUT" DISTANCE IN KILOMETRES ";DO
180 IFDO=0ORDO>999THENPRINT@160,SP$:GOTO170
190 :
200 PRINT:PRINTMI$:PRINT
210 LO=100*PO/CO:KO=DO/LO:MO=2.82*KO:L1=100*LO/DO
220 PRINT" NO. OF LITRES BOUGHT  = ";USING F1$;LO
230 PRINT" KILOMETRES PER LITRE  = ";USING F1$;KO
240 PRINT" MILES PER GALLON      = ";USING F1$;MO
250 PRINT" LITRES/100 KILOMETRES = ";USING F1$;L1
260 PRINT:PRINTMI$
290 :
300 PRINT@485,"LIKE TO TRY AGAIN  Y/N";:SOUND30,1
310 IN$="":IN$=INKEY$:IN$=INKEY$:IFIN$=""THEN310ELSESOUND20,1
320 IFIN$="Y"THEN100ELSEIFIN$="N"THENCLS:ENDELSE310
390 :
400 '**********************************************************
410 '* - - - - - - - I N S T R U C T I O N S - - - - - - - *
420 '* ---------------------------------------------------- *
430 '* PRICE OF PETROL BOUGHT ? - ENTER SUM PAID AT SERVICE *
440 '* STATION IN DOLLARS AND CENTS - IE :- 28.00           *
450 '* ---------------------------------------------------- *
460 '* PRICE/LITRE IN CENTS   ?  -    IE :- 57.90           *
470 '* ---------------------------------------------------- *
480 '* DISTANCE IN KILOMETRES ? - ENTER DISTANCE TRAVELLED  *
490 '* SINCE YOU LAST FILLED UP YOUR PETROL TANK. IE:- 318.4 *
500 '* ---------------------------------------------------- *
510 '* ONCE YOU PRESS RETURN ON YOUR LAST QUESTION, DETAILS *
520 '* OF YOUR RUNNING EXPENSES WILL BE DISPLAYED ON SCREEN. *
530 '* ---------------------------------------------------- *
540 '* - - - - - - H A P P Y - M O T O R I N G - - - - - - *
550 '**********************************************************
560 '* NOTE -:- DO NOT TYPE IN LINES 10 TO 50 AND 390 TO 570 *
570 '**********************************************************
```

FIXING AC PLUG PACK ADAPTOR :-

This started to cut out at embarrasing moments. Moving the cord near where it exited from the case 'CURED' it, but any bump cut power. I was unwilling to 'RETURN TO HONG KONG' and operating on the principle what one man puts together, another can take it apart.

I seperated the case sections where they join in the middle. Tools used - Sharp cutting knife and electric screw driver, plus a fair bit of pushing and shoving. It seemed to work so I put it back together. (Re-glued with Supa-Glue, BIG MISTAKE). It still cut out so removed case again.

Noticed that where the cord exits there are plastic lugs to hold it firmly. Where these had pressed, the cord was flattened. Removed the cord ends and cut through the cord, past the flattened section. Resoldered the new ends, used soldering iron to melt part of restrainig lug. I put a blob of silastic where the cord exits and resealed case, using different glue and electrical tape. The end result isn't neat but it works.

EDITORS COMMENT :-

For anyone contemplating repairing their own AC Plug Adaptors I would strongly advise them to leave it to the experts. 240V AC is LETHAL. Unless you know what your'e doing leave it alone. We prefer you among the living.

For persons insisting on repairing their AC Plug Adaptors I would like to point out that there's another way to take the case apart, but don't forget you are doing it at your own risk.

1) Put plug pack on it's side on a hard surface like a breadboard.

2) With a small hammer tap along the joint repeating on all four sides till the glue cracks. Be carefull not to hit too hard as it will damage the case. Too soft and glue will not crack.

The two places where the lead usually breaks is where it leaves the case and near the plug as well. When repairs are done reassemble using a glue designed for plastics as the wrong type could damage case.

FIND UPDATE BY ROBERT QUINN . . .

I modified the FIND routine along the line of Larry Taylor's ENHANCED FIND, making it work properly with LPRINT command as suggested by the editor in issue #15 of the H.V. VZ Journal.

Also line numbers and memory locations are now seperated with COLONS instead of spaces. The mods are very simple, line numbers are as in Larry's ASSEMBLER LISTING in the Journal :-

```
076     CALL 032AH
079     LD   A,3AH
080     CALL 032AH
090     CALL 032AH
```

DOT MATRIX GRAPHICS by Larry Taylor

Whilst most EPSON printers and their innumerable emulators share a common set of print codes, there are often extensive variations to each printer command set. Competing brands, in an attempt to outdo the oppposition and distinguish themselves from other printers in their price range, will offer various enhancements over a base EPSON model. These may include a choice of typefaces and printstyles, letter quality print, or even increased speed. Some of these improvements are achieved through hardware modifications, such as increasing the number of pins in the printhead, whilst others are due to the software installed in the printer's ROM.

One area where software differences between printers can be immediately noticed, is in the way a single character is printed on paper. Each character printed, consists of columns of dots in series. The number of columns, and in particular the number of dots in each column, will vary according to the capabilities of the printhead in each printer. Software can, however, make a difference to the final appearance of the character. My first EPSON compatible printer was a BMC BX-80, which eventually gave way to an OLYMPIA NP165. A simple comparison of characters printed by these two machines showed there was a very obvious difference. The dot columns printed by the BMC are close together, whilst those displayed by the OLYMPIA are slightly spaced. This results in a character printed by the BMC being marginally narrower than that created by the OLYMPIA. This difference in dot density, is due to the control, which the ROM software exercises over the data, which ultimately operates the printhead. Most EPSON type printers possess two, bit image modes, that allow the user to directly control, which pins are fired in the printhead. All the characters produced by a printer are created using these bit image facilities. However, because the printer stores the machine code driver routines and the character shape tables (these are sets of numbers, which determine a character's shape) in its own ROM, it can churn out its own predesigned shapes considerably faster, than it can the individual bit image data transmitted to it by a host computer. The use of an inbuilt RAM buffer and a facility for downloading character data to the printer is an attempt to partly overcome the speed limitations of dot image processing.

The term, bit image, may need some explanation. A byte of data on a computer, such as the VZ, consists of eight bits. Each of these individual bits can only be equal to one or zero. To correspond to this, a printhead, in its simplest form, consists of 8 pins arranged vertically. When a single byte of data is sent to the printhead, the value of the individual bits will determine which pins are fired. For example, if bit 4 is equal to one then pin 4 will produce a dot on the paper. This is often referred to as the printer's graphics mode, because it can be used to produce shapes drawn by the user. The size and shape of the image to be produced is limited only by the width of paper the printer can handle.

The two, bit image modes on most EPSON type printers are usually referred to as single or normal density and double density. These terms, basically refer to the number of dots that can be printed across the page. In double density mode, twice the number of dots can be printed across the page as there would be in single density, however, since the paper width is the same, the dots have to be much closer together, giving the resulting image a much darker appearance. The actual number of dots printed varies from printer to printer. The EPSON standard usually specifies 480 dots in single density mode and 960 dots in double density. My BMC BX-80, however, does not conform to this and chooses, instead, to use 640 dots and 1280 dots respectively. The OLYMPIA, on the other hand, whilst conforming to the EPSON original, additionally offers, a further four 8 pin bit image modes and two, which utilise a ninth pin. One of the 8 pin modes is referred to as quadruple density and can print 1920 dots across the page. As well, this printer allows for 640 dot and 1280 dot modes to be accessed. These modes are cryptically referred to as CRT graphics, which I can only assume stands for Cathode Ray Tube, since the manual doesn't tell me. Not being familiar with the screen resolution of oscilloscopes, I can only guess that these dot widths are necessary to allow the full width of such images to be dumped to the printer. Although these modes haven't been assigned the standard EPSON codes, it does show that the OLYMPIA can emulate the BMC, if necessary.

This lack of conformity is what caused problems for me, when I came to write the screen dump routines for the VZ-EPSON Printer Patch. As we all know, the MODE(1) screen, on the VZ, is 128 pixels wide and there can only be four colours on the screen at any one time. To print a representation of the screen on paper, it is necessary to first work out a dot image to stand for each differently coloured pixel. Green was most easily represented as blank, whilst red was displayed as completely black. Yellow had to be shown as a slightly lighter image than that displayed for blue. I then had to decide on how many dots wide, I would make each pixel image. To calculate this, the number of pixels, 128, is multiplied by the number of dots in each printed image and then displayed as a total. The simple mathematics is displayed below.

128 X 3 dots = 384 dots
128 X 4 dots = 512 dots

I was unaware of the differences between printers outlined above, so in my earlier versions of the patch I opted for a 4 dot width, which meant that the screen dump was 512 dots wide. Since my BMC was capable of a 640 dot width and I was well within that, I experienced no problems. Not so, those purchasers of the patch who owned EPSON type printers, which conformed to the 480 dot standard. These people discovered, to their dismay, as I did after buying the OLYMPIA, that when they did a screen dump, 32 dots or 8 pixels from the right hand side of their picture didn't fit on the page. It was therefore necessary to make a small change to later versions of the patch (Version 1.4 being the most recent), so that the pixel images used were only 3 dots wide. This gave dumps with a total dot width of 384 dots. Solving this problem was easy, but because of the difference in dot density between the BMC and OLYMPIA, screen dumps done on the BMC now appear too narrow. I cannot see any simple solution to this problem, as it is controlled by the software within each printer's ROM. Nevertheless, I hope that this has helped to answer some of the queries I have had from people conmcerning operation of the screen dump.

In this next section, I will attempt to explain how the standard bit image modes of an EPSON compatible printer can

be accessed from BASIC. Because the method used for either mode is very similar, our discussion will focus on single density dot graphics. Various printer manuals display the necessary code in one or more of the following ways.

ASCII	ESC	K	+ n1	+ n2
Decimal	(27)D	K(75)D	n1	n2
	$[27]_{10}$	$[75]_{10}$	[n1]	[n2]
Hexidecimal	(1B)H	(4B)H	(n1)	(n2)
	$[1B]_{16}$	$[4B]_{16}$	[n1]	[n2]

The first two values [27] and [75] put the printer into normal density bit image mode. The next two values, n1 and n2 tell the printer how many bytes of graphic data will be transmitted. They are arranged in low byte, high byte order. For example if we were doing a screen dump, each line would require 384 bytes of data. The low byte and high byte can be calculated in this way.

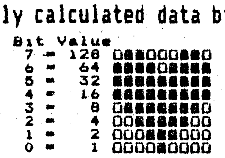
high byte n2 = INT(384 / 256)
low byte n1 = 384 - n2 * 256

Using this method n1 = 128 and n2 = 1. These values will, of course vary according to the number of bytes to be sent. What follows next, is a stream of graphics data, which, in the case of the example used above, will consist of 384 individual bytes. Provided the low byte, high byte values, when calculated out,

eg. no. of data bytes = n1 + 256 * n2

equals the number of graphics data bytes which follow, the design will be printed. If the printer does not respond by printing the graphic, it will usually mean that either the low byte - high byte values haven't been entered correctly, or that the printer has not yet received the expected number of data bytes and is waiting for them.

To illustrate the procedure, I will demonstrate how to print this heart shaped character ♥ using single density, bit image, dot graphics. This character consists of nine dot columns. Each column is represented by a single byte of data. In addition, every column is made up of eight rows, each of which is represented by a single bit within the byte of data. When reading a column from the bottom up we begin at bit 0, which has the value 1, and finish at bit 7, which is equal to 128. A single data byte is calculated by totalling these row values assigned to each of the dots printed in a column. Because the character consists of nine columns, it is represented by nine similarly calculated data bytes.



By thus totalling each column in turn, vertically, from left to right we arrive at the following byte values.

112, 248, 252, 126, 63, 126, 252, 248, 112

The number of data bytes to be sent to the printer to produce this particular character is nine. Represented in low byte - high byte form this is shown as,

low byte = 9
high byte = 0

So to produce the heart character, we send the following data to the printer.

Put printer in graphics mode :- 27,75,
Number data bytes (low byte - high byte) :- 9,0,
Data bytes :- 112,248,252,126,63,126,252,248,112

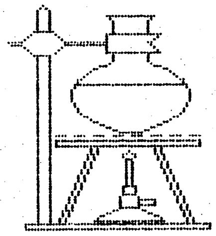
The easiest way to accomplish this from BASIC, would be as follows.

```
100 REM SEND EPSON CODE TO PUT PRINTER IN SINGLE DENSITY BIT
110 REM IMAGE MODE.
115 REM ***************************************************
120     LPRINT CHR$(27);"K";
125 REM ***************************************************
130 REM LOW BYTE - HIGH BYTE DATA TO TELL PRINTER HOW MANY
140 REM BYTES OF GRAPHICS DATA TO EXPECT.
145 REM ***************************************************
150     DATA 9,0
155 REM ***************************************************
160 REM NINE GRAPHICS DATA BYTES NECESSARY TO PRODUCE A
170 REM HEART SHAPED CHARACTER.
175 REM ***************************************************
180     DATA 112,248,252,126,63,126,252,248,112
185 REM ***************************************************
190 REM SOME VALUES SUCH AS 0,10,11,12,13 AND THE INVERSE
200 REM AND GRAPHICS CHARACTER VALUES FROM 128 TO 255 WON'T
210 REM REACH THE PRINTER IF WE LPRINT THEM, SO SEND THE LOW
220 REM BYTE - HIGH BYTE AND NINE DATA BYTES OUT THE PORTS.
225 REM ***************************************************
230     FOR T=1 TO 11:READ D: GOSUB 300:NEXT
240     END
245 REM ***************************************************
250 REM READ PORT ZERO TO CHECK IF PRINTER IS READY. WHEN
260 REM IT IS, SEND DATA OUT PORTS 13 AND 14.
265 REM ***************************************************
300     IF (INP(0)AND1)<>0 THEN 300 ELSE OUT 13,D:OUT 14,D
310     RETURN
320 REM ***************************************************
```

To further assist those of you, who may still be experiencing difficulty in getting your EPSON type printer to produce dot graphics, I have written a short DOT MATRIX GRAPHICS EDITOR in BASIC. This program enables a design to be drawn on the screen by moving a cursor within a grid, which is 8 rows high and 11 columns wide. Once the design is completed and the RETURN key pressed, the graphics data is automatically tabulated by the program. Following this, the printer is set to single density bit image mode and the eleven bytes of graphics data are sent to the printer, preceded, of course by the low byte - high byte equivalent of the number eleven.

Once the concept of producing images on the printer, using dot graphics, has been fully grasped and you have experienced the first joys of success, there remains only to experiment with your own designs. There are two ways, in which you can create your images prior to transferring them to the printer. The first is to do a drawing on squared paper, calculate the column values, create data statements and then transmit them to the printer. The second, possibly easier method, depending on the resolution required, is to use an EDITOR to draw on the HIRES screen. On completion, it can read the screen, compute the data and send it to your printer. Having used both approaches, I find an EDITOR easier. Don't despair if you don't experience immediate success, my early frustrations culminated in the successful creation of the VZ-EPSON Printer Patch.

```
100 REM ################################################
110 REM # DOT MATRIX GRAPHICS EDITOR BY LARRY TAYLOR  #
120 REM ################################################
130 REM
150 REM ################################################
160 REM # CONTROL KEYS FOR DOT MATRIX GRAPHICS EDITOR #
170 REM ################################################
180 REM # [M] - CURSOR LEFT        [,] - CURSOR RIGHT #
200 REM # [.] - CURSOR UP      [SPACE] - CURSOR DOWN  #
240 REM # [L] - INSERT BLOCK       [;] - RUBOUT BLOCK #
250 REM # [RETURN]-PRINT SHAPE     [C] - CLEAR SCREEN #
290 REM ################################################
300 REM # INITIALISE VARIABLES                        #
310 REM ################################################
320 P=28672:D=60:E=207:DIMV(10)
330 CLS:POKE30744,1:COLOR2
340 REM ################################################
350 REM # SET UP EDITOR DISPLAY                       #
360 REM ################################################
370 PRINT@  0,"▛▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▜";
380 PRINT@ 32,"▌  DOT MATRIX GRAPHICS EDITOR  ▐";
390 PRINT@ 64,"▙▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▟";
400 FORT=3TO14:POKEP+T*32,154:POKEP+T*32+31,149:NEXT
410 PRINT@480,"▙▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄";:POKEP+511,151
420 P=28844:PRINT@130,"BIT VALUE▄▄▄▄▄▄▄▄▄▄▄▄▄"
430 FOR T=7 TO 0STEP-1
440 PRINT@162+(7-T)*32,USING"##  ";T;:PRINTUSING"### ";2^T;
450 PRINT" ▐           ▌":NEXT
460 PRINT@427,"▀▀▀▀▀▀▀▀▀▀▀▀▀"
470 M=X+Y*32:V=PEEK(P+M)
480 REM ####################################
490 REM # READ KEYPRESS AND FLASH CURSOR #
500 REM ####################################
510 A$=INKEY$:POKEP+M,V:FORT=1TOD:NEXT:POKEP+M,E
520 FORT=1TOD:NEXT:A$=INKEY$:IFA$=""THEN510
530 A=ASC(A$):IFA=67THENRUN
540 IFA=13THEN560
550 IFA<>32ANDA<>44ANDA<>46ANDA<>59ANDA<>76ANDA<>77THEN510
560 POKEP+M,V:FORT=1TOD:NEXT
570 IFA$=CHR$(13)THENGOSUB690
580 IFA$="M":X=X-1:IFX<0 :X=0
590 IFA$=",":X=X+1:IFX>10:X=10
600 IFA$=".":Y=Y-1:IFY<0 :Y=0
610 IFA$=" ":Y=Y+1:IFY>7 :Y=7
620 M=X+Y*32
630 IFA$="L":POKEP+M,128
640 IFA$=";":POKEP+M,96
650 V=PEEK(P+M):GOTO510
660 REM ##########################
670 REM # CALCULATE DATA VALUES #
680 REM ##########################
690 FORX=0TO10:FORY=0TO7:M=X+(7-Y)*32
700 IFPEEK(P+M)<>128THEN720
710 V(X)=V(X)+2^Y
720 NEXT:NEXT
730 REM ##########################
740 REM #  PRINT GRAPHIC IMAGE  #
750 REM ##########################
760 LPRINTCHR$(27);CHR$(75);
770 D=11:GOSUB790:D=0:GOSUB790
780 FORT=0TO10:D=V(T):V(T)=0:GOSUB790:NEXT:X=0:Y=0:RETURN
790 IF(INP(0)AND1)<>0THEN790ELSEOUT13,D:OUT14,D:RETURN
```



```
5 DIMD(772)
8 LPRINTCHR$(27);"A";CHR$(7);
10 FORT=1TO772:READD:D(T)=D:NEXT
15 FORI=0TO11:LPRINT"  ";
18 FORX=1TO4:D=D(X):GOSUB50:NEXT
20 FORT=1TO64
25 D=D(4+I*64+T):GOSUB50:NEXT
35 LPRINT:NEXT:END
50 IF(INP(0)AND1)<>0THEN50ELSEOUT13,D:OUT14,D:RETURN
90 DATA27,75,64,0
200 DATA0,0,0,0,0,0,0,0,63,64,128,64,63,0,0,0,0,0,0,0,0,0,0,0
205 DATA0,0,0,0,0,8
210 DATA12,11,8,8,8,8,8,8,8,8,8,8,11,12,8,0,0,0,0,0,0,0,0,0,0
215 DATA0,0,0,0,0,0
220 DATA0,1,1,1,1,2,4,8,16,240,16,16,16,240,16,8,4,2,1,1,1,1,1
225 DATA1,1,1,1,1,1,1,1,15
230 DATA8,248,8,8,8,8,8,8,8,8,8,8,248,9,10,12,8,0,0,0,0,0,0,0,0
235 DATA0,0,0,0,0,0
240 DATA128,64,64,64,192,32,16,8,4,7,4,4,4,7,4,8,16,32,192,64,64
245 DATA64,64,64,64,64,64,64,64,64,64,248
250 DATA8,15,8,8,8,8,8,8,8,8,8,8,143,72,40,24,8,0,0,0,0,0,0,0,0
255 DATA0,0,0,0,0,0
260 DATA0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,0,0,0,1,2,4
265 DATA8,16,16,32,32,32
270 DATA64,128,0,0,0,0,0,0,0,0,0,128,64,32,32,32,16,16,8,4,2,1
275 DATA0,0,0,0,0,0,0,0
280 DATA0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,15,48,224,32
285 DATA32,32,32,32,32,32,32,32
290 DATA32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32,32
295 DATA32,32,32,32,224,48,15,0,0,0,0,0,0
300 DATA0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,192,48,12,2
305 DATA1,0,0,0,0,0,0,0
310 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,12,48,192
315 DATA0,0,0,0,0,0
320 DATA0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,0,0,0,0
325 DATA128,64,32,32,16,16,8
330 DATA8,4,4,2,2,2,2,2,2,2,2,4,4,8,8,16,16,32,32,64,128,0,0,0,0
335 DATA0,0,0,0,0,0
340 DATA0,0,0,0,0,0,0,0,255,0,0,0,255,0,184,168,168,168,40,40
345 DATA168,168,168,168,40,47,168,175,168,168,40
350 DATA40,168,168,168,168,41,42,169,168,168,168,40,40,168,175
355 DATA168,175,40,40,168,168,168,168,40,40,168,168,168,184,0,0
356 DATA0
360 DATA0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,0
365 DATA0,0,3,28,227,28,224,0,0,0,0
370 DATA0,0,0,0,192,63,32,63,192,0,0,0,0,0,0,224,28,227,28,3,0,0
375 DATA0,0,0,0,0,0,0,0,0
380 DATA0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,0,0,1
385 DATA14,113,142,112,128,0,0,0,0,0,0
390 DATA0,0,0,0,7,252,4,252,7,0,0,0,0,0,0,0,0,128,112,142,113,14
395 DATA1,0,0,0,0,0,0,0,0,0
400 DATA0,0,0,0,0,0,0,0,255,0,0,0,255,0,0,0,0,7,56,199,56,192
405 DATA0,0,0,0,0,0,0,0,0
410 DATA0,0,1,62,194,66,66,66,194,62,21,20,20,20,28,0,0,0,0,0
415 DATA192,56,199,56,7,0,0,0,0,0,0,0
420 DATA0,0,0,0,0,7,5,5,5,253,5,5,5,253,5,5,29,229,29,229,5,5,5
425 DATA5,5,5,5,13,29,45,45,77
430 DATA77,141,13,13,13,13,13,13,13,13,13,141,77,77,45,45,29,13
435 DATA5,5,5,5,5,229,29,229,29,5,5,5,5,7
```

As I was getting quite a few disk I/O errors I decided to give the head a clean using a CLEANMASTER head cleaning disk purchased from Dick Smith. The head is nice and clean now, but my drive no longer READS or WRITES anymore and I have'nt had chance to fix the problem as I had a deadline to meet with this issue. I suspect the pressure pad may have been damaged. My second drive is OK and I'm thankful it did'nt suffer the same fate as it's head was cleaned first.

The reason for the mishap is quite simple, it's lack of proper instructions that come with the disk. The cleaning disk is designed for dual head drives and as the VZ drive has only a single head it's quite easy to put it in the wrong way and that's what I did. The instructions say to insert the C/Disk according to the arrows, but there's no arrows on disk. For VZ drives use the following steps :-

1) Turn C/Disk over and put no more than 5 (FIVE) drops of cleaning fluid to both oblong cutouts on back of disk.

2) Turn disk right way up again so that the label with instructions is on top.

3) Turn disk 90 Deg. so that the label is on the left side of disk and the WRITE PROTECT notch is away from you.

4) Insert C/Disk in drive and close drive door.

5) Type in DIR and press RETURN. This will start the cleaning process.

6) After 30 Sec's press (-) key to stop drive. Open drive door and remove C/Disk.

Below are a series of drawings showing the CLEANMASTER disk for persons not familiar how the C/Disk is used by the drive.

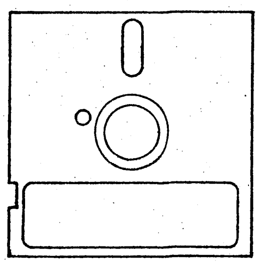
The drawing on the left depicts top side of C/Disk while the one next to it shows reverse side of C/Disk. Note the three cutouts on this side.

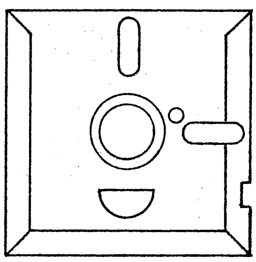
The third drawing shows the drive recording head, inserted disk and pressure pad. With the drive door open the pressure pad is lifted up allowing for easy removing/inserting of disks.

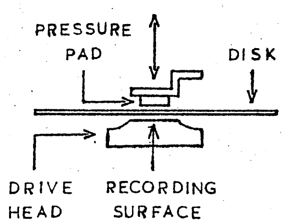
The C/Disk must be inserted into drive as shown in the last drawing on the right. The reason for that is because the C/Disk is made of different material and has a rougher surface. If inserted the normal way the pressure pad would rest on the cleaning surface and there would be a good possibility of the pressure pad tearing the cleaning surface which is what happened to me.

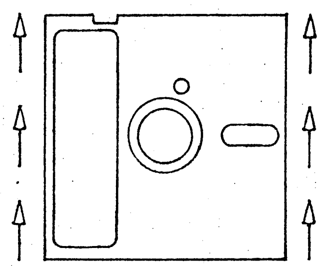
The act of closing drive door lowers pressure pad onto the exposed surface of the disk in the cutout pressing disk against the head where reading/writing from/to disk is done. Note :- READING/WRITING is done on bottom of disk.

The comments above relate to the CLEANMASTER HEAD C/DISK only. If using other brands of C/Disk make sure that the cleaning surface is on bottom and that the pressure pad does not rest on any part of cleaning surface.









GALACTIC EMPIRES
SOFTWARE WRITTEN AND MARKETED BY SCOTT LE BRUN
AND REVIEWED BY PETER J HILL (AUCKLAND N.Z)

This program includes a very comprehensive and easy to understand scenario. The instructions are also easy to follow. So read them and play--You will be glad you did!.

You-The power hungry emperor of a small star system ,must build up your empire and dispatch your star ship's to explore other star systems.

The object of all this exploring is to plunder and conquer these star systems and thereby increase your power and wealth to continue feeding your lust for supremacy.

You can have up to 4 players in this game or you can play the master gamer (YOUR VZ). You can also nominate how many turns you wish to play, thus avoiding a never ending game and no result. You can even rename your home star, if you wish. (Note new name must start with F).

Once you have fed the VZ this data, you will be informed that the computer is working thing's out. This is a really nice touch and at least you don't start thinking the games hung up when faced with a blank screen.

By pressing the 0 key you will be given a command list/menu which will enable you to gather information, such as battle results, star ship locotion, star map etc.

Galactic Empires is a super space simulation where you must pit your skill and cunning to beat your opponent, who is out to protect his own planets. I feel that Galactic Empires is as good as, if not better than the star trek game which is available.

You will enjoy playing this game and it will really go to your head when you start building your empire. Scott has done a good job writing this program, as he has done with many others.

This game is a must for the adventurous gamer and is well worth the cost, which at the time of writing this review is only A$12-50. TOP MARKS FOR THIS ONE SCOTT

AVAILABLE FROM :- Scott LE BRUN 59 BRENTWOOD DRIVE
WANTIRNA VICTORIA 3152 AUSTRALIA

FOR PRIVATE SALE . . . .

```
7A2B:3AFB68     LD   A,(68FB)
7A2E:FE73       CP   73
7A30:CC477A     CALL Z,7A47
7A33:FE79       CP   79
7A35:CC437A     CALL Z,7A43              INTERRUPT ROUTINE
7A38:FE5B       CP   5B
7A3A:CC437A     CALL Z,7A43
7A3D:FE7A       CP   7A
7A3F:CC437A     CALL Z,7A43
7A42:C9         RET

7A43:320070     LD   (7000),A
7A46:C9         RET

7A47:2A2078     LD   HL,(7820)
7A4A:010072     LD   BC,7200
7A4D:3A1878     LD   A,(7818)
7A50:FE00       CP   00
7A52:3E20       LD   A,20
7A54:2002       JR   NZ,02 (7A58)
7A56:3E60       LD   A,60
7A58:77         LD   (HL),A              CURSOR CLEAR
7A59:323C78     LD   (783C),A            SCREEN ROUTINE
7A5C:23         INC  HL
7A5D:79         LD   A,C
7A5E:BD         CP   L
7A5F:20EC       JR   NZ,EC (7A4D)
7A61:78         LD   A,B
7A62:BC         CP   H
7A63:20E8       JR   NZ,E8 (7A4D)
7A65:C9         RET

7A66:F5         PUSH AF
7A67:C5         PUSH BC
7A68:D5         PUSH DE
7A69:E5         PUSH HL
7A6A:3E00       LD   A,00
7A6C:329C78     LD   (789C),A            DIR/STATUS
7A6F:CD0649     CALL 4906                ROUTINE
7A72:CDCD52     CALL 52CD
7A75:E1         POP  HL
7A76:D1         POP  DE
7A77:C1         POP  BC
7A78:F1         POP  AF
7A79:C9         RET

7A7A:F5         PUSH AF
7A7B:C5         PUSH BC
7A7C:D5         PUSH DE
7A7D:E5         PUSH HL
7A7E:CDC405     CALL 05C4
7A81:CB47       BIT  0,A                 LDIR/LSTATUS
                                         ROUTINE
7A83:2007       JR   NZ,07 (7A8C)
7A85:3E01       LD   A,01
7A87:329C78     LD   (789C),A
7A8A:18E3       JR   E3    (7A6F)
```

```
7A8C:21E600    LD   HL,00E6
7A8F:014B00    LD   BC,004B
7A92:CD5C34    CALL 345C
7A95:E1        POP  HL          LDIR/LSTATUS
7A96:D1        POP  DE          ROUTINE CONT.
7A97:C1        POP  BC
7A98:F1        POP  AF
7A99:C9        RET
```

The first part of the disassembly is for an INTERRUPT routine to scan for pressing of SHIFT X or SHIFT C or SHIFT V or SHIFT B keys. The INTERRUPT routine is set up in a block of unused memory in the communications region, starting at address 31275.

Three of the CALLs in this routine, one for each pair of keys excepting SHIFT C, address the start of the same short routine, also set up in the same block of memory after the end of the INTERRUPT routine. This CALLed routine copies the content of the A register into the first cell of video memory whenever one of the three pairs of keys are pressed.

To set up and activate the INTERRUPT and various CALLed routines, use the accompanying INTDATA program.

If you now hold down the SHIFT key and press <X> or <V> or <B>, you will see a character appear in upper right of screen, a different character for each key.

Each of the four CALLs in the INTERRUPT routine can be changed to address any routine in ROM or RAM; that routine can then be executed when the allocated pair of keys are pressed. The address is POKEd into the CALL instruction in LO HI format :

```
HI%=ADDRESS/256:LO=ADDRESS-HI%*256:PRINTLO,HI%
```

LO is POKEd in as the second byte of the CALL, HI% as the third byte (see the disassembly).

When run, the INTDATA program sets up a CURSOR CLEAR SCREEN (CCS) machine code routine (31303 to 31333). This routine allows you to blank out the screen from the cursor position to bottom right of screen but leaving intact the content of the screen to the left of and above the cursor.

SHIFT C keys have been allocated to this function. The LO and HI of the start address of CCS are 71 and 122. INTER has POKEd these two numbers into the SHIFT C CALL address in the INTERRUPT.

Move the cursor to any position on the screen, hold down SHIFT and press C key to clear screen from cursor position. If your screen is full of text and other garbage, you can enter commands like LIST, RUN, PRINT, etc., without having to move the cursor to bottom of screen, by using CCS to selectively wipe clean the screen. Because CCS is accessed thru the INTERRUPT EXIT, it can be used anytime, even if a BASIC program is running.

For those of you who have a disk drive, there is a DIR:STATUS routine in the code set up by INTDATA, starting at 31334. To allocate SHIFT B keys to this routine in the INTERRUPT,

POKE31296,102

Now (making sure you have disk in drive and door is shut) you can display the directory and status of the disk anytime (even if a BASIC program is running) simply by holding SHIFT and pressing B key.

There is also included in the code an LPRINT DIR:STATUS routine. To allocate SHIFT V keys to this routine,

POKE31291,122

Again, with your drive ready, hold SHIFT and press V key. You hear a warning sound only? You must have forgotten to switch on your printer! With printer on, hold SHIFT and press V key to get a print out of the directory and status of the disk in the drive.

I leave SHIFT X to your imagination. What do YOU want it to do?

Disk drive owners can BSAVE the INTERRUPT and associated routines thus :

BSAVE"INTER",7A2B,7A99

To use from disk, BLOAD"INTER" then activate the INTERRUPT by POKEing a jump to start of INTERRUPT (JP 31275) into the INTERRUPT EXIT at 30845,

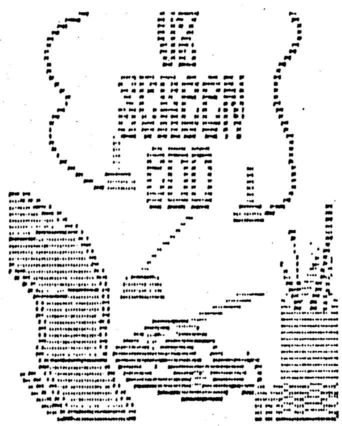
POKE30846,43:POKE30847,122:POKE 30845,195

Be sure you POKE 30845 last otherwise you will activate the INTERRUPT EXIT before you have POKEd the address you want it to jump to.

Any routine you design to be executed as a function of these keys should start by PUSHing onto the STACK any registers used in the routine and end by POPing those registers back from the STACK.

```
2 '*****************************************************
4 '*  INTERRUPT FUNCTION KEYS DESIGNED BY ROBERT QUINN  *
6 '*****************************************************
8 :
10 FORR=31275TO31385:READA:POKER,A:B=B+A:NEXT
20 IFB<>13921THENSOUND30,2;20,1:PRINT"[illegible]":END
30 POKE30846,43:POKE30847,122:POKE30845,195
100 DATA58,251,104,254,115,204,71,122,254,121,204,67,122
120 DATA254,91,204,67,122,254,122,204,67,122,201,50,0,112,201
130 DATA42,32,120,1,0,114,58,24,120,254,0,62,32,32,2,62,96,119
140 DATA50,60,120,35,121,189,32,236,120,188,32,232,201
150 DATA245,197,213,229,62,0,50,156,120,205,6,73,205,205,82,225
160 DATA209,193,241,201
170 DATA245,197,213,229,205,196,5,203,71,32,7,62,1,50,156,120,24
180 DATA227,33,230,0,1,75,0,205,92,52,225,209,193,241,201
```

MTS PRESENTS



VZ Screen Edd is a new powerful utility program for the VZ computers. The utility allows easy and fast drawing of high resolution screens for use in what ever you please.

The program is fitted out with a whole host of features including instant backgrounds, built in lettering, many types of printer dumps, a fully transperant cursor, fast screen movement and the option of using joystick. The utility is also fitted out with both disk and tape functions and is standard for any set up as it makes system checks before trying anything that may not be connected such as a disk drive.

The utility's most useful feature though is its ability to both load and save to tape. No screen editor has previously been able to do so in a sucessful and usable way. Here begins the world of screen shows for tape owners. Yes, as described in the manual it is possible to make a screen show on tape.

The utility comes complete with a six page manual which also contains a video worksheet for your own use and on top of that there is also four example screens to get you started. The manual fully explains how to use the utility, how to use your designed screens and how to go about designing a screen from start to finish. A full list of commands is also in the manual. The utility also, because of its setup, is able copy screens easily between tape and disk and between tape and tape etc.

Overall this utility presents a very worth while investment for both tape and disk owners. Disk owners can also use the tape facility as a reliable back up for their screens. Who knows what someone will dream up as a usage for it.

PRICE $25.00 + $2.00 P&H

See me at a H.V.VZ.U.G meeting and pay only $25.00 or $20.00 if you supply a tape or disk or,

SEND YOUR CHEQUE OR MONEY ORDER TO :-

Matthew Taylor
38 Fishing Point Road,
RATHMINES NSW 2283 AUSTRALIA

(049) 752350

MTS
MATTHEW TAYLOR SOFTWARE

EXTENDED DOS VERSION 1.0 (C) - COPYRIGHT - DAVE MITCHELL - 1987

MERGE - MERGES basic file from disk with program in memory.
DIRA - See example - T:MENU B:PATCH3.1 B:WORDPROC
B:EXTDOS E B:EXTDOS R W:DOS-INST
LDIRA - As above, but to screen and printer.
DIRB - See example - T:MENU 01 00 7AE9 801B 0532
B:PATCH3.1 01 0B 7200 771F 051F
LDIRB - As above, but to screen and printer.
STATUSA - Prints free disk space to screen on one line.
LSTATUSA - As above, but to screen and printer, see below.
534 RECORDS FREE 63.500K FREE
OLD - Restores a program after using the NEW command.
OLD. - Prints START, END and LENGHT of program in memory in HEX.
DEC XXXXX - Converts DECIMAL to HEX
HEX XXXX - Converts HEX to DECIMAL
STATUSA and LSTATUSA also works with Version 1.0 DOS.

The EXTENDED DOS is available in the two versions below :-
EXTDOS R - T.O.M. SEEKING (SELF RELOCATING)
EXTDOS E - FOR 2K RAM AT 6000-67FF HEX
Price - $10.00 each or the two for $15.00. Availble from :-

Dave MITCHELL - (079) 27 8519
24 ELPHINSTONE STREET NORTH ROCKHAMPTON QUEENSLAND 4701

FOR INFORMATION IN NEWCASTLE AREA :- Joe LEON - (049) 51 2756

FOR SALE - E & F W.P. PATCH 3.1

PATCH3.1 - COPYRIGHT - HUNTER VALLEY VZ USERS' GROUP

This single Patch will convert your E & F TAPE WORD PROCESSOR for full DISK use while retaining all TAPE functions. It can be used with 1 or 2 DRIVES. Below are the two Menus.

E)DIT TEXT	L)OAD
C)LEAR TEXT	S)AVE
P)RINT TEXT	D)IR
L)OAD FILE	E)RA
S)AVE FILE	R)EN
V)ERIFY FILE	I)NIT
Q)UIT PROGRAM	1-2) DRIVE 1
D)ISK	M)ENU

Fast SAVING and LOADING of TEXT DATA to and from Disk is provided using Block SAVE or LOAD.

Full instructions are supplied together with a Tape to Disk transfer utility for your E & F Tape Word Processor.

This Patch will work with V1.0 or V1.2 Disk Controller. A STATUS facility has been added for V1.0 DOS owners.

SYSTEM REQUIREMENTS :- DISK DRIVE + V1.0 OR V1.2 DOS
VZ300 + 16K RAM PACK OR VZ200 + 18K (16K RAM PACK + 2K)

The price - $10.00, NZ AU$12.00 and is available from :-

HUNTER VALLEY VZ USERS' GROUP
P.O.BOX 161 JESMOND 2299
N.S.W. AUSTRALIA.Phone (049)51 2756